

<only if time is highlighted blue>

<assigned reviewers are highlighted yellow>

Who is here: Chencheng Li, Zack, Zhenyu, Haoting, Pranay, Ryan Garcia, JayLynne Redeaux, Nick, Katy

Overall goal: Build game logic loop, MVP, home page & collection if time

Dates: Team calendar

- Logic due by: 5/29 (please give testing team a bit of a buffer; have at least 85% done by 5/29)
- Testing/review due by: 6/1

Team 1: Game Logic <SPRINT 3> (Nick, Darwin, Katy)

- Make sure the game logic works with the HTML
- Fix the timer
- Fix the AI card slot
- Make sure the winning condition works in the end
- Communicate in depth with Team 2

Team 2: Game Pages Layout <SPRINT 3> (Zack, Destin, JayLynne, Zhenyu, Nick)

- Talk more to teams 1/2
- Make sure the cards display properly
- Make sure the timer link to the JS from Team 1
- Winning/lose pop up
- Card winning counter on the top right
- Create img for the card

Team 3: Custom element, card description, collection page <SPRINT 3> (Ryan, Chengcheng, Pranay, Zack, Haoting)

- Add absolute path
- Put all img into one folder
- Resize images (and have ChatGPT edit the images to follow the game theme more?)
- Change the fix css card size to dynamically
- User collections, user active deck in localStorage
 - User collection -> List of keys
 - User active deck -> subset of the user collection
 - Before the game starts, we randomize the user active deck List.randomize(), pop from the top to draw

- When we need to render the cards on the page, then we search DB by key to get the card object

Team 4: CI Pipeline/Testing (Chung Lee, Haoting, Pranay)

- Unit tests for IndexedDB and localStorage
 - **5 per each** (User collection -> List of keys
User active deck -> subset of the user collection
Before the game starts, we randomize the user active deck List.randomize(), pop from the top to draw)
- Puppeteer for game-page.html (Toward the end)
 - Make the puppeteer to play the game(keep choosing cards until the game ends, check if the winner is correct)
- ESLint for inline CSS and JS
- GitHub pages deployment from develop branch
 - Play the game on the live side (everyone can play the game)
 - Set up the github page correctly
- Lighthouse on GitHub pages
 - Main develop – fancy site (only for main)
 - Sandbox develop – github page
 - Can only do develop branch

Local built

Guidelines: general rules / reminders for Sprint 3

- General workflow for development:
 - feature branches should be branched off of develop. Make sure to periodically pull from develop.
 - Run ESLint and prettier locally:


```
npm run lint
npm run format
npx prettier . --check
npx prettier . --write
```
 - (if applicable) run the unit tests: `npm run test:unit`
 - (if applicable) run the e2e tests: `npm run test:e2e`
 - When ready to merge to develop, make a PR request. Make sure you pass the CI pipeline. Tag your assigned reviewer at least in your PR.

- Post in #stand-up every other day. Use #breaking-changes for changes that are made on your end that will affect other teams. Use threads for replies; keep the general channel clear.
- Use Github Projects (we will close the previous sprints'). The general project for your team and <SPRINT 3> will be created, but you must create sub-issues and correctly assign yourself to them. Keep your Github Project board updated, and link PRs by commenting closes #<issue number>.
- Please acknowledge Slack messages, even in #important-announcements

Deadlines:

- 05/29: Midterm & at least 85% of logic done.
- 6/1: admin/meetings/060125-retrospective.md → Leaders will compile our retrospective materials and submit this once we meet.
- 6/1: admin/meetings/060125-sprint-review-3.md → Leaders will compile our sprint review materials and submit this once we meet.